Integrating Plagiarism Detection into a Class Management System for Academic Integrity

Neda Lotfi
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

Suhair Amer
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

Ankita Maharjan
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

ABSTRACT
This system is a comprehensive database designed to assist faculty members in efficiently managing academic information and monitoring student performance. It allows tracking and maintenance of key data, including class details, student records, activities, grades, and submitted documents. Faculty members can store and update information such as student names and addresses, class names, numbers, locations, times, and enrolled students. The system also supports management of classroom activities by recording activity names, numbers, maximum points, due dates, and instructions. A built-in grade book enables faculty to maintain and analyze student scores for each activity, while an integrated plagiarism detection feature compares newly uploaded files against previously submitted ones to ensure academic integrity. Additionally, the system provides tools for retrieving information, identifying missing assignments, and generating performance summaries such as highest, lowest, and average scores, thereby enhancing the overall efficiency of academic record management.

Keywords
Database system, student management, class information, grade book, plagiarism detection, academic records, faculty tools, performance tracking, information retrieval, automation

1. INTRODUCTION
With the advancement of digital age, education has evolved beyond traditional classes. Teachers can now effectively manage class, track students' progress and distribute educational material easily with technology driven platform like class management systems. The learning management system (LMS) is a software based on a web server, cloud computing or personal local computer that manages the teaching and learning process in an academic or non-academic program without the constraint of time and place [1].

However, academic dishonesty and plagiarism have also increased as online materials have become more widely available. Plagiarism is presenting someone else's work as one's own word without any acknowledgement or citation. It is a serious academic offence as it violates honesty, integrity, and originality in scholar work. As internet resources become more widely available and AI writing tools proliferate, the likelihood of both inadvertent and deliberate plagiarism has significantly increased. As a result, plagiarism detection has emerged as a crucial phase in the academic writing process, assisting researchers, teachers, and students in avoiding the moral and professional consequences of unintentional duplication [2].

To identify the best LMS platform for both educators and learners, the University at Buffalo recently conducted an evaluation of three popular platforms: Canvas, Brightspace, and Blackboard Ultra [3]. Usability, proficiency in technology, system integration, and stakeholder input, including surveys and practical testing, were all considered during the assessment. In addition to providing integration with plagiarism detection systems to uphold academic integrity, the results showed that Canvas was the most popular platform because of its user-friendly interface, extensive instructional features, and excellent alignment with institutional needs. This highlights the vital role that plagiarism detection and efficient LMS platforms play in improving learning outcomes and moral academic conduct.

Abdulquadir et al. [4] conducted a usability study on the Canvas Learning Management System to evaluate its effectiveness in supporting online class activities. According to their research, Canvas frequently encountered problems because of its complicated interface. These results suggest the importance of interface design and usability when building a class management system. Along with it, it is also equally important to incorporate a plagiarism detection tool guaranteeing both effective course administration and advancement of academic integrity.

Liu et al. [5] introduce DWright, a tutorial system that teaches paraphrase and correct citation to assist students write in English without plagiarizing. According to their evaluation, users considered the tool's interface to be useful and improved their understanding of proper citation formats and paraphrasing techniques. The authors emphasized that plagiarism prevention can be improved by combining usability design with instructional guidance and scaffolding.

To address the growing problem of academic dishonesty in digital learning environments, Arkan and Abdul [6] worked on improving the Moodle Learning Management System by creating and including an automated plagiarism detection function. Testing revealed that the system was error-free and completely compatible with Moodle, proving its viability and usefulness in educational contexts. Their main objective was to demonstrate the benefit and integration possibilities of plagiarism detection in Moodle. Similarly, Tresnawati et al. [7] developed a plagiarism detection tool within Moodle, addressing challenges such as diverse coding styles and false positives. By automatically comparing student code contributions to identify possible duplication, their approach demonstrated how LMS platforms may be successfully expanded to assist plagiarism prevention in both textual and code-based evaluations. Their system could display the proportion of similarity between student assignments, both individually and across numerous files, which made it easier for teachers to identify possible instances of duplicated code [7].

The selection of academic integrity tools has been significantly increased by the integration of AI and machine learning. A machine learning-based model was created by Volarić, Martinović, and Ljubić [8] to identify academic fraud in online exams, particularly during the COVID-19 pandemic when remote assessments was popular. Their method collected human–computer interaction (HCI) data, including keystrokes and mouse activity, from 54 participants using a browser plugin called Student Activity Tracker, producing over 500,000 logs. Strong relationships between user behavior and fraudulent activity were found by the investigation, proving that behavioral analytics and machine learning can detect cheating in ways other than text comparison. In class management systems, this method signifies a move away from conventional plagiarism detection and toward real-time behavioral monitoring and predictive analytics [8].

Similarly, Crossley et al. [9] combined traditional text-based similarity metrics with an analysis of keystroke dynamics, or the time and rhythm of students' typing, to present a novel method for detecting plagiarism. Their technique separates genuine from duplicated text by combining linguistic aspects with biometric and interaction data (such typing speed, pauses, and revision patterns). The study showed that including human–computer interaction (HCI) data greatly increases the precision of plagiarism detection, particularly for language that has been paraphrased or helped by artificial intelligence. The authors concluded that to develop more dependable, customized, and invincible models of academic integrity monitoring, future detection systems must incorporate behavioral analytics and content analysis [9].

The expansion of AI in education has also raised new challenges for existing plagiarism detection software. Research by Mohammad and Erkan, explored how effective plagiarism detection tools worked on ChatGPT and other AI-generated works. According to their research, ChatGPT can generate complex and incredibly "original" writing that frequently avoids being picked up by traditional plagiarism detection software. They discussed the need for institutions to consider appropriate measures to mitigate potential plagiarism issues and advise on the ongoing debate surrounding the impact of AI technology on education [10]. This is particularly important now because integrity tools and class management systems need to change to recognize, comprehend, and react to content that was not intended for use by previous platforms.

Vandenhoek evaluated two popular plagiarism detection programs, Grammarly and Google Classroom Originality Reports, identified problematic material in student essays. The study evaluated the percentage of highlighted text that truly represented possible plagiarism using samples from Japanese university students. In his conclusion, Vandenhoek emphasized the significance of instructor supervision, critical examination of results, and enhanced instructional use of such technology in academic settings. He said that plagiarism detection programs should be seen as helpful tools rather than definitive authority [11].

The increasing practical and ethical issues surrounding the use of AI detection tools in educational settings have been investigated by Hirsch [12]. Although the purpose of these systems is to detect writing produced by artificial intelligence, they frequently generate false positives that unfairly single out students with unique writing styles and non-native English speakers. Hirsch points out that bias, poor judgment, and a decline in student trust might result from these detectors' unclear text classification process. Instead of depending exclusively on automated results, the article advises educators to use AI detection tools with caution and responsibility, applying human judgment and contextual understanding. Hirsch concludes that to balance equity, respect for students' rights, and proper application of technology in the classroom, academic integrity policies need to change.

A thorough survey evaluating eight plagiarism detection algorithms in Arabic, French, and English was carried out by Abdelhamid, Azouaou, and Batata [13]. Their study assessed the algorithms' capacity to identify different forms of plagiarism, such as cross-language translation, paraphrase, and direct copying. Along with outlining the shortcomings of current systems, they examined plagiarism categories and key detection methods such string matching, fingerprinting, and semantic similarity. The authors noted that linguistic and translation complications make multilingual detection particularly difficult, and they concluded that future systems should concentrate on creating language-independent and semantically aware methods.

To meet the changing demands of digital education, Belani et al. [14] developed and put into use an improved LMS. By adding real-time communication features, customized scheduling, and more robust authentication techniques like biometric login, their suggested approach enhances conventional LMS systems. Utilizing ISO 25010 software-quality standards, the system prioritizes security, performance, and usability essential characteristics that guarantee dependable academic operations. The report emphasizes how collaborative, adaptable, and integrity-focused capabilities are necessary for LMS platforms to transcend beyond content management. While boosting accessibility and participation is the paper's primary focus, it also acknowledges plagiarism detection and data protection as crucial elements for future advancement, highlighting the importance of developing thorough and secure class management systems in modern education [14].

The future of class management systems depends on striking a balance between academic integrity and technological innovation, according to Qazi et al. [15] and Fidas et al.[16]. According to Qazi et al., CMS will develop more adaptive and moral learning environments in the future by combining automation, artificial intelligence, and academic integrity protections. According to their research, to improve educational quality and equity, contemporary CMS platforms need to include behavioral monitoring, plagiarism detection, and tailored learning analytics . This is in accordance with the overall pattern of research on plagiarism detection and class management systems, which indicates that in an increasingly digital age, the next generation of CMS tools must enhance academic integrity and institutional credibility in addition to improving usability and learning outcomes [15][16].

2. PROJECT DESCRIPTION AND TOOLS

This system is a database for tracking information such as classes information, students' information, store student activities, maintain grades, and detect plagiarism. This allows a faculty member to maintain: student information such as name, address, class information such as class name, number, location, time, and list of students enrolled, activities such as activity name and number, maximum points, due date, instructions, student grade book for each activity scores/grades and submitted documents.
In specific, the faculty will be able to:

- Add information
- Delete information

- Update information
- Retrieve information
- Check for plagiarism.
- Check who has missing assignments
- Check who has the highest score
- Check who has the lowest score.
- Check what the average score is.

Next are the tools, languages and libraries used to implement the system.
Project Language: JAVA
Tools: NetBeans IDE 8.2, MySQL 8.0
Library:

- Mysql Connector Java 0.0.27, JDK 1.8
- Commons compress 1.19
- Poi 4.1.1
- Poi ooxml 4.1.1
- Poi ooxml schemas 4.1.1
- Poi scatchpad 4.1.1
- Xmlbeans 3.1.0

3. BENEFITS OF DEVELOPING THIS SYSTEM

Developing this system offers several significant benefits for both faculty and academic institutions. It centralizes all class-related data, making it easier to manage student information, class schedules, grades, and activities from a single platform. The automation of tasks such as grade calculation, plagiarism detection, and assignment tracking saves time and reduces the likelihood of human error. Faculty members can quickly identify students with missing assignments or low performance, allowing for timely interventions and improved student outcomes. The system also enhances academic integrity through its built-in plagiarism checking feature. Additionally, it promotes transparency and consistency in grading, simplifies data retrieval for reports or evaluations, and improves overall communication and organization within the educational environment.

Developing a Class Management System with Plagiarism Detection offers numerous benefits to students. It provides valuable hands-on experience in software development, database management, and system integration while enhancing technical skills in programming, web design, and the use of plagiarism detection algorithms. Through this project, students also gain a deeper understanding of academic integrity and the importance of originality in academic work. Additionally, it fosters teamwork, project management, and problem-solving abilities—key skills for professional growth. Beyond technical learning, the project contributes to education by creating a tool that supports instructors in managing classes efficiently and encourages students to submit authentic, plagiarism-free work.

4. METHODOLOGY

The system was developed using a structured methodology that included requirements specification, system design, database construction, implementation, and testing. The goal was to create a database-driven application that enables faculty to manage student records, class information, activities, grades, and plagiarism detection.

In the requirements specification phase, functional requirements were identified to support the management of:

- Student information (name, address, contact).
- Class information (course name, number, time, location, roster).
- Activities (activity name/number, maximum points, due date, instructions).
- Grades and submitted files. Additionally, the system must allow the faculty to add, update, delete, and retrieve information, detect plagiarism, identify missing assignments, and compute score statistics (highest, lowest, average).

For system design, the system adopts a three-tier architecture consisting of:

- Presentation Layer: Java-based GUI developed in NetBeans.
- Business Logic Layer: Java classes implementing CRUD operations, grade calculations, and plagiarism analysis.
- Data Layer: MySQL 8.0 database storing all entities, including students, classes, enrollments, activities, and grades.

A normalized database schema with appropriate foreign keys and indexes was designed to ensure data integrity and efficient querying.

The system was implemented in Java (JDK 1.8) using NetBeans IDE 8.2. MySQL Connector/J enabled communication with the database. CRUD functionalities were implemented with prepared statements for secure and efficient SQL execution. Activity management included storing points, deadlines, and instructions, while the gradebook module calculated missing assignments, highest/lowest scores, and averages using SQL aggregate functions.

For plagiarism detection, document submissions were processed using the following libraries:

- Apache POI (poi, poi-ooxml, poi-schemas, scratchpad)
- Commons Compress
- XmlBeans

Submitted files were converted to text and compared using n-gram and similarity-based algorithms to identify potential plagiarism.

For integration of libraries, external libraries were integrated through NetBeans project dependencies. POI handled file extraction, Commons Compress supported compressed formats, and XmlBeans enabled structured document parsing.

Testing included:

- Unit testing of Java modules.
- Database testing for CRUD accuracy and referential integrity.
- Functional testing of grade calculations, plagiarism detection, and reporting features.
- User acceptance testing to evaluate usability and completeness.

Finally for deployment, the application was packaged with all required libraries and deployed as a standalone Java program connected to a MySQL database via configuration settings.

5. DESIGN, IMPLEMENTATION AND TESTING

The design and implementation of the system were centered around creating a user-friendly, efficient, and reliable database for managing academic information. The system architecture follows a modular design, dividing functionality into key components such as student management, class management, activity tracking, grade book management, and plagiarism detection. The database was structured using relational database principles to ensure data integrity and efficient retrieval, with tables representing entities like students, classes, activities, and submissions connected through primary and foreign keys. During implementation, a suitable database management system (DBMS) that is MySQL was used to store and organize the data. The front-end interface was designed to allow faculty members to easily add, update, delete, and retrieve information through simple forms and menus. The backend logic was implemented using the Java programming language, which handled user requests, data validation, and database interactions. For plagiarism detection, the system incorporated text comparison algorithms that analyze similarities between newly uploaded and previously submitted documents. Overall, the design emphasizes scalability, data accuracy, and ease of use, while the implementation ensures smooth integration between different modules, providing faculty with a powerful tool to manage and monitor student performance effectively.

Figure 1 shows the main menu of the system. When clicking on 'Add student' button, the add new student menu [Figure 2] will allow the user to enter the required information and then click save button. The student information will be updated in the DB [Figure 3]. The user can click on 'Student Information' button the main menu to display student information [Figure 4]. On the right of each student information, the use can click 'Action button' that will take them to the Edit/delete menu [Figure 5] and they can choose either to edit or delete the student information.

User can edit the student information through the Edit Student menu [Figure 6] and click the save button. They can also add a new class through the Add New Class menu [Figure 7] and click save button and that will updated the DB [Figure 8]. User can display the class information [Figure 9] and can click the action button for any class where they can then choose to edit, delete or show students [Figure 10]. If they choose to show the students, a list will be displayed [Figure 11]. User an edit a class [Figure 12], add a student to a class [Figure 13], which will update the DB [Figure 14], display students in a class [Figure 15].

The user can choose to add a new activity, such as an assignment, quiz, report, lab, etc. using the add new activity window [Figure 16] which will update the DB [Figure 17] and then can view it [Figure 18] and make changes to a specific activity by clicking on the action button. The user can choose to edit or delete an activity [Figure 19] and [Figure 20].

The user can grade an activity and submit the grade [Figure 21] which will updated the DB [Figure 22]. Finally, the user can view who received the highest score [Figure 23], who received the lowest score [Figure 24], and the average score [Figure 25].

To check the plagiarism of an activity, the user will use the simple plagiarism cheder window [Figure 26], which will ask the user to select a file [Figure 27] and will display results [Figure 28]

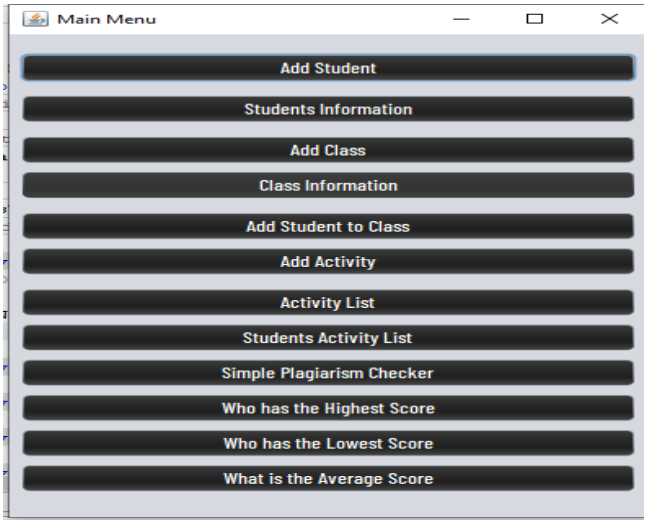


Fig 1: Main menu

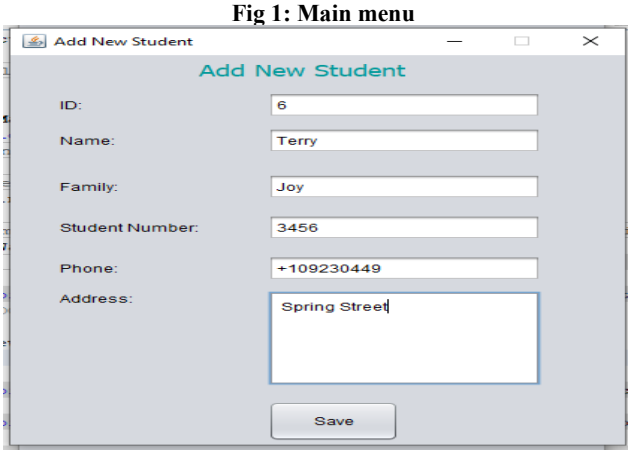


Fig 2: Add a new student menu

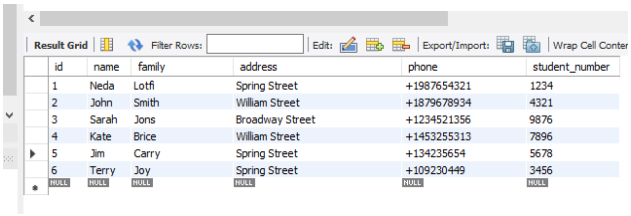


Fig 3: Student added to DB

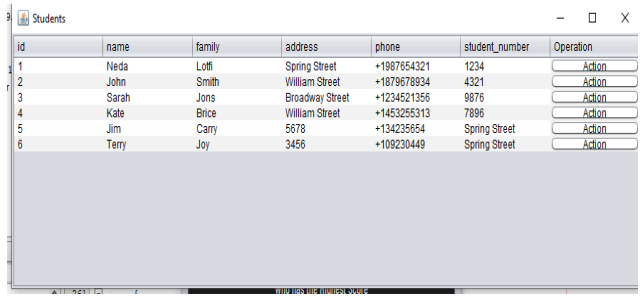


Fig 4: Display student information form/menu

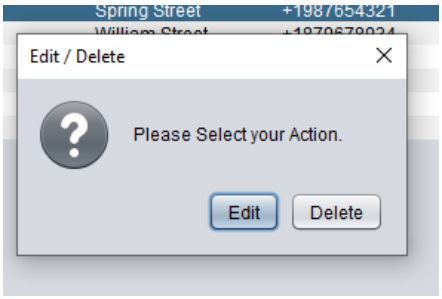


Fig 5: Edit/delete student information menu

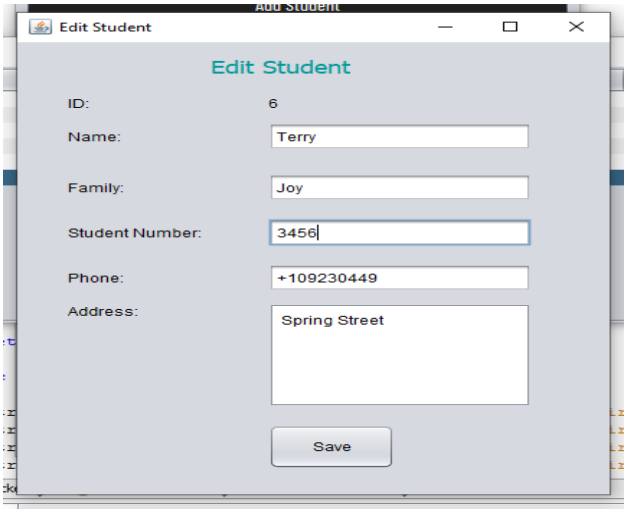


Fig 6: Edit student menu

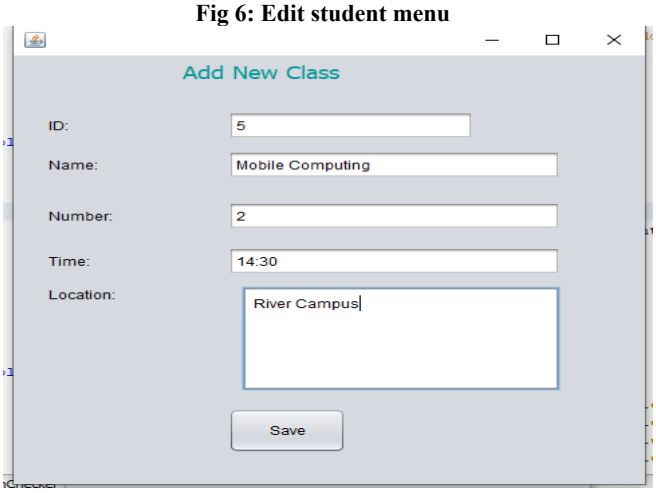
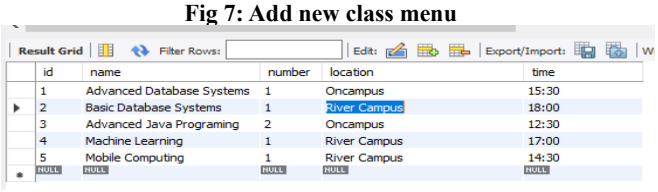


Fig 7: Add new class menu



id	name	number	location	time
1	Advanced Database Systems	1	Oncampus	15:30
2	Basic Database Systems	1	River Campus	18:00
3	Advanced Java Programing	2	Oncampus	12:30
4	Machine Learning	1	River Campus	17:00
5	Mobile Computing	1	River Campus	14:30

Fig 8: class added to the DB

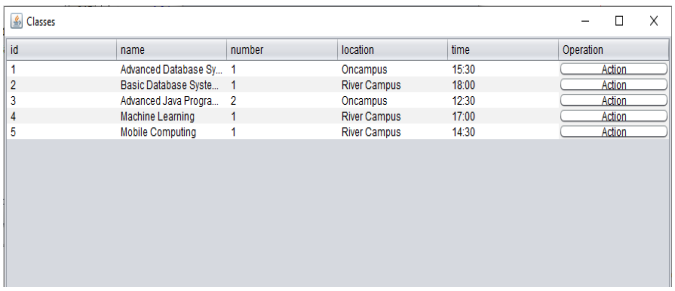


Fig 9: display class information form/menu

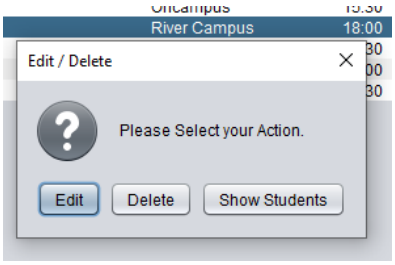


Fig 10: Edit/delete/show students class information menu

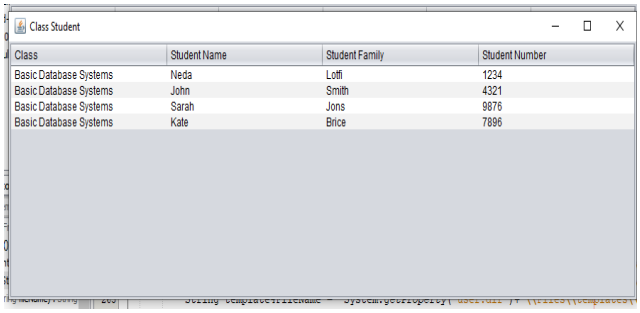


Fig 11: Show students form

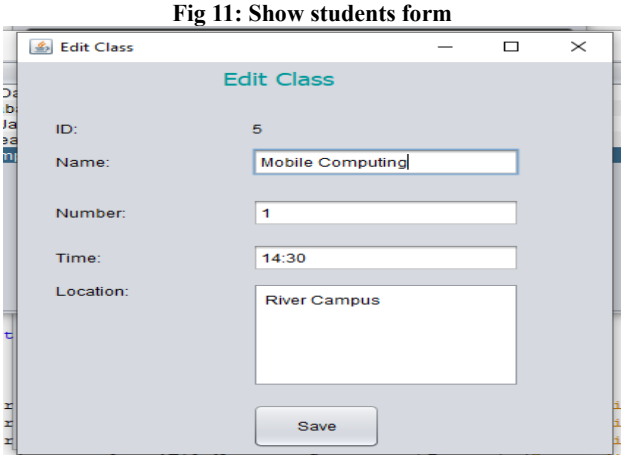


Fig 12: Edit class menu

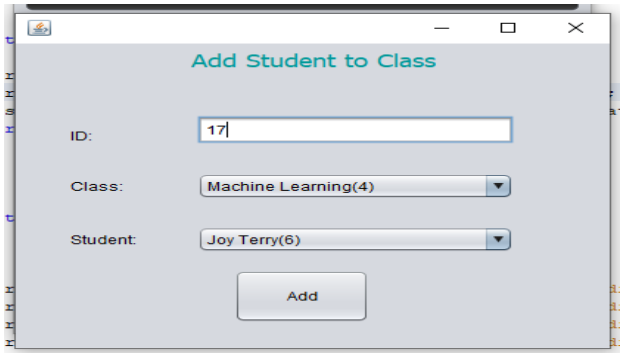


Fig 13: add student to class menu

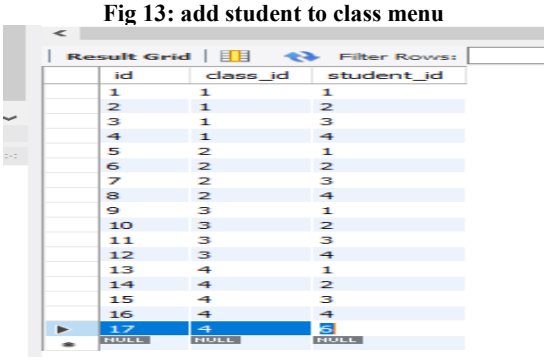


Fig 14: added student to class in DB

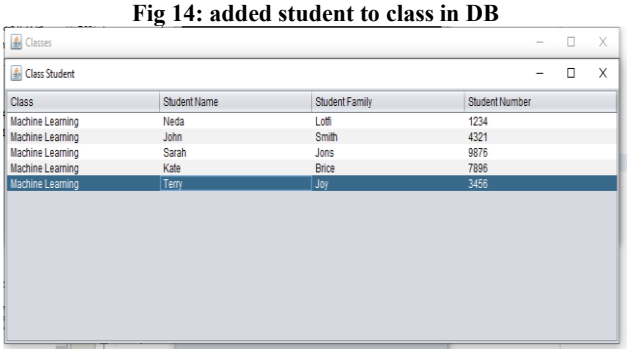


Fig 15: displaying students enrolled in a specific class.

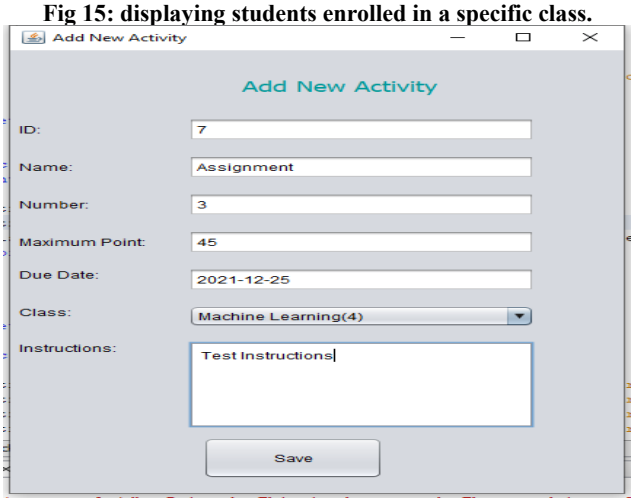


Fig 16: add a new activity menu

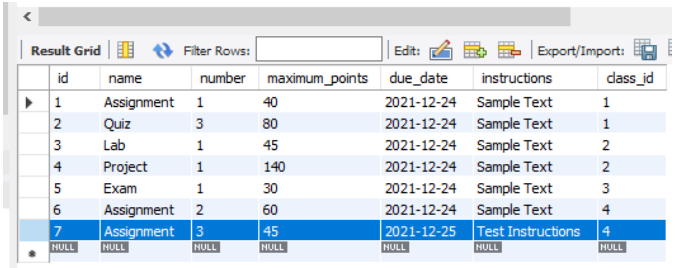


Fig 17: adding the activity to the DB

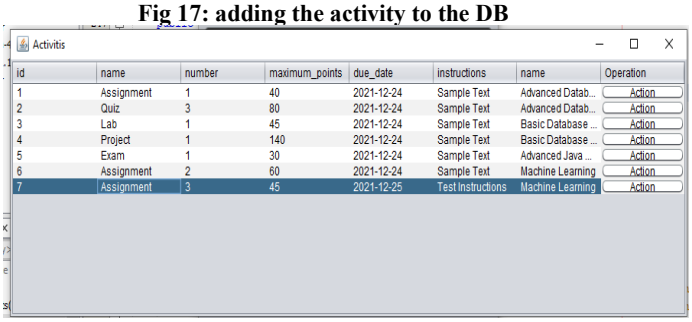


Fig 18: displaying activity menu.

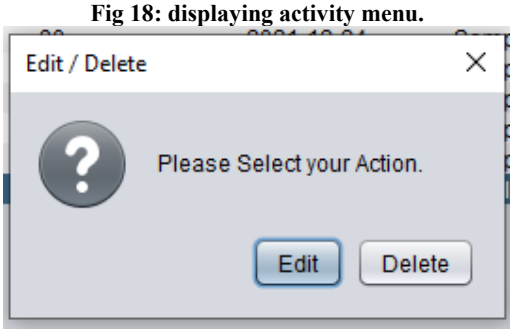


Fig 19: Edit/delete an activity menu

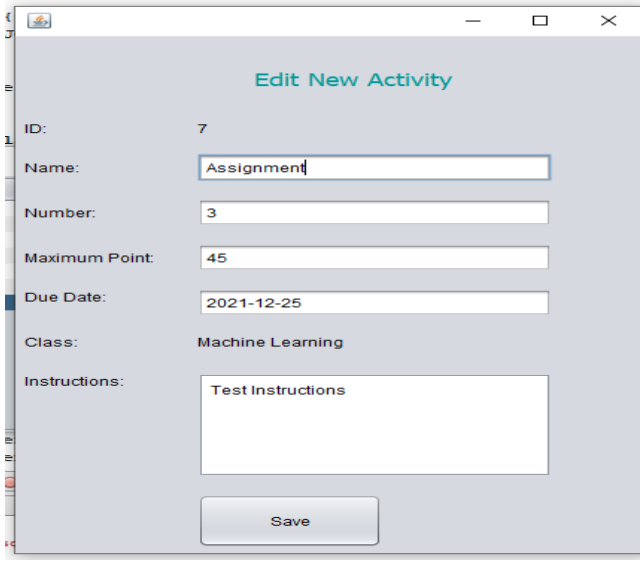


Fig 20: Edit an activity menu

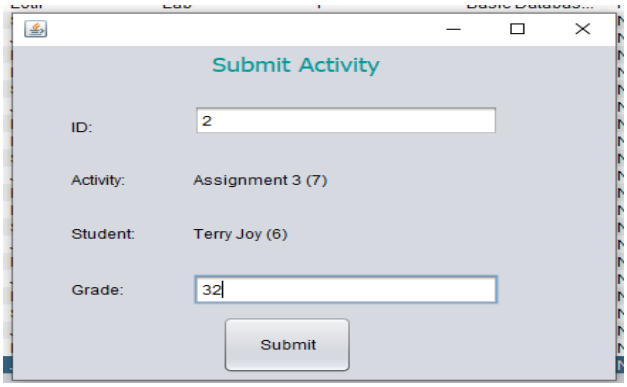


Fig 21: submit a grade menu

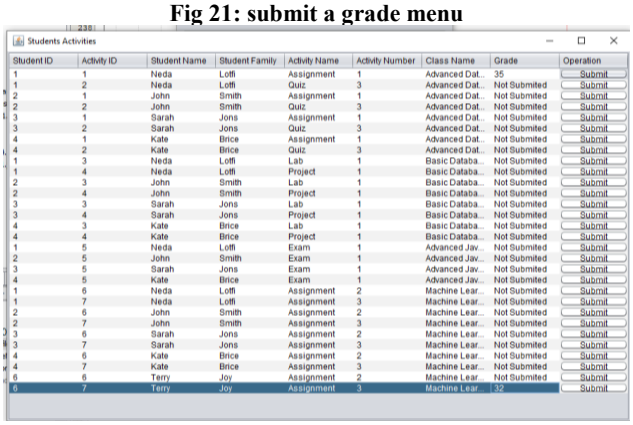


Fig 22: grade updated in DB

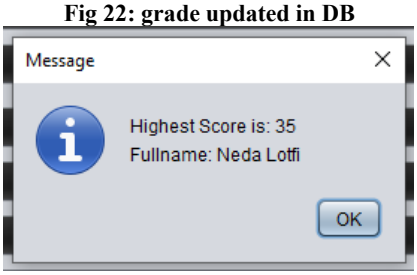


Fig 23: displaying highest score

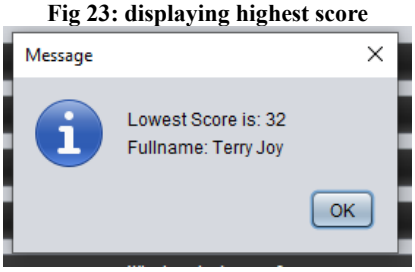


Fig 24: displaying lowest score

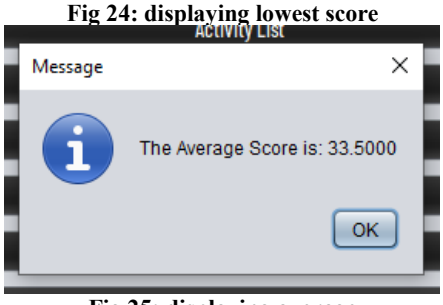


Fig 25: displaying average

We used a set of ten reference documents to test whether an eleventh document contained plagiarism. The ten documents served as the original sources for comparison, representing different writing styles and content areas. By analyzing the similarities between the eleventh document and these reference texts, we were able to identify overlapping phrases, sentence structures, and ideas that might indicate copied or reused material. This approach allowed us to evaluate the effectiveness of our plagiarism detection method and determine if the eleventh document was original or had significant content borrowed from the others.

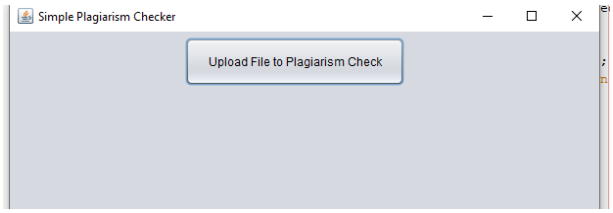


Fig 26: uploading a file to check for plagiarism menu

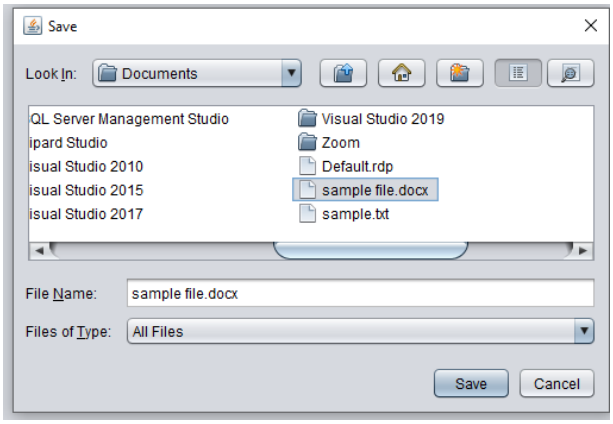


Fig 27: selecting a file.

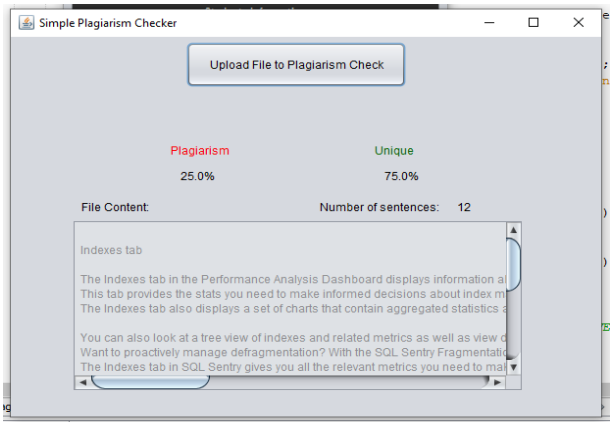


Fig 28: displaying plagiarism status

6. EVALUATION

The system was evaluated based on functionality, performance, accuracy, and usability. All core features—data management, grade calculations, missing assignment detection, and plagiarism checking—performed correctly during testing. Database queries and file-processing tasks executed efficiently with no noticeable delays. Accuracy checks confirmed that score calculations matched expected values and that the plagiarism module correctly identified similar submissions. Usability testing with sample users showed

that the interface is clear and easy to navigate. Error handling was effective, with appropriate messages displayed for invalid inputs or connection issues. Overall, the system met the project requirements and operated reliably under typical usage conditions.

7. CONCLUSION

In conclusion, this database system provides an effective solution for managing and monitoring all aspects of classroom information in a structured and efficient manner. It streamlines faculty tasks by centralizing student records, class details, activities, and grades in one platform. The inclusion of features such as plagiarism detection, performance tracking, and automated calculations of averages and scores enhances academic integrity and supports informed decision-making. By enabling easy addition, deletion, updating, and retrieval of information, the system not only saves time but also improves accuracy and consistency in record keeping. Overall, it serves as a valuable tool for faculty members to maintain organized data, promote transparency, and improve the teaching and learning process.

Future improvements to the Class Management System with Plagiarism Detection can focus on enhancing functionality, accuracy, and user experience. Advanced artificial intelligence and natural language processing (NLP) techniques could be integrated to improve plagiarism detection by identifying paraphrased or translated text more effectively. Expanding the system to support multiple file formats and programming languages would increase its flexibility for diverse academic fields. Incorporating real-time plagiarism feedback could help students revise their work before final submission, promoting learning and originality. Additionally, developing a mobile-friendly version or cloud-based deployment would improve accessibility and scalability. Integration with institutional databases and learning management platforms could further streamline academic workflows and promote seamless adoption in educational institutions.

8. REFERENCES

[1] Sanchez, L., Penarreta, J., & Soria Poma, X. (2024). Learning management systems for higher education: A brief comparison. Discover Education, 3(1). https://doi.org/10.1007/s44217-024-00143-5

[2] León, J. L., Quaresma, P., & Nogueira, V. (2024, October). *Plagiarism detection: Report.* University of Évora.

[3] LMS Review Committee. (2022, January 11). *LMS Review Report and Recommendations*. University at Buffalo. https://www.buffalo.edu/content/dam/www/lms/project-overview/LMS_Review_Report_and_Recommendations_011122.pdf

[4] Ahmad Ayobami Abdulquadir, Opeyemi Abdulsalam, Russell Mcmahon, and Annu Prabhakar. 2023. Usability Study of a Learning Management System (LMS). In Proceedings of the 24th Annual Conference on Information Technology Education (SIGITE '23). Association for Computing Machinery, New York, NY, USA, 150–156. https://doi.org/10.1145/3585059.3611415

[5] Gi-Zen Liu, Hsiang-Yee Lo, Hei-Chia Wang. Design and usability testing of a learning and plagiarism avoidance tutorial system for paraphrasing and citing in English: A case study, Computers & Education, Volume 69, 2013, Pages 1-14, ISSN 0360-1315, https://doi.org/10.1016/j.compedu.2013.06.011.

[6] A. K. S. Sabonchi and A. K. Görür, "Plagiarism detection in learning management system," *2017 8th International Conference on Information Technology (ICIT)*, Amman, Jordan, 2017, pp. 495-500, doi: 10.1109/ICITECH.2017.8080048.

[7] Dewi Tresnawati, Arief Syaichu R, Kuspriyanto. Plagiarism Detection System Design for Programming Assignment in Virtual Classroom based on Moodle, Procedia - Social and Behavioral Sciences, Volume 67, 2012, Pages 114-122, ISSN 1877-0428, https://doi.org/10.1016/j.sbspro.2012.11.312.

[8] Volaric, Tomislav & Martinovic, Goran & Ljubić, Hrvoje. (2023). Detecting Academic Fraud at Online Tests During COVID-19 Using Machine Learning-Based Methods. 10.1007/978-3-031-36833-2_11.

[9] Scott Crossley, Yu Tian, Joon Suh Choi, Langdon Holmes, & Wesley Morris. (2024). Plagiarism Detection Using Keystroke Logs. Proceedings of the 17th International Conference on Educational Data Mining, 476--483. https://doi.org/10.5281/zenodo.12729864

[10] Khalil, M., Er, E. (2023). Will ChatGPT Get You Caught? Rethinking of Plagiarism Detection. In: Zaphiris, P., Ioannou, A. (eds) Learning and Collaboration Technologies. HCII 2023. Lecture Notes in Computer Science, vol 14040. Springer, Cham. https://doi.org/10.1007/978-3-031-34411-4_32

[11] Vandenhoek, T. (2023, December 31). Assessing the Accuracy of Plagiarism Detection Systems. International Journal of Education and Development using ICT [Online], 19(3). Available: http://ijedict.dec.uwi.edu/viewarticle.php?id=3209.

[12] Hirsch, A. (2024, December 12). *AI detectors: An ethical minefield*. Center for Innovative Teaching and Learning. https://citl.news.niu.edu/2024/12/12/ai-detectors-an-ethical-minefield/citl.news.niu.edu

[13] Abdelhamid, M., Azouaou, F., & Batata, S. (2022). *A survey of plagiarism detection systems: Case of use with English, French and Arabic languages.* arXiv preprint arXiv:2203.00871. https://arxiv.org/abs/2203.00871

[14] Belani, A. J., Bahri, A. R. D., Izhar, A., Shadman, B., Mukhtiyar, F., Das, P., & Amin, N. U. (2025). *Design and implementation of an enhanced learning management system: Addressing modern academic challenges.* Preprints. https://doi.org/10.20944/preprints202505.1522.v1

[15] Sameer Qazi, Muhammad Bilal Kadri, Muhammad Naveed, Bilal A. Khawaja, Sohaib Zia Khan, Muhammad Mansoor Alam, Mazliham Mohd Su'ud. *AI-driven learning management systems: Modern developments, challenges, and future trends during the age of ChatGPT. Computers, Materials and Continua, Volume 80, Issue 2, 2024, Pages 3289-3314, ISSN 1546-2218,*https://doi.org/10.3991/ijet.v19i6.40217

[16] Fidas CA, Belk M, Constantinides A, Portugal D, Martins P, Pietron AM, Pitsillides A, Avouris N. Ensuring Academic Integrity and Trust in Online Learning Environments: A Longitudinal Study of an AI-Centered Proctoring System in Tertiary Educational Institutions. *Education Sciences*. 2023; 13(6):566. https://doi.org/10.3390/educsci13060566